Malaysian Science and Technology Congress, Melaka, 2001

Neural Networks for Process Monitoring, Control and Fault Detection: Application to Tennessee Eastman Plant

Arshad Ahmad [a], Mohd Kamaruddin Abd Hamid [*]

Center for Process Simulation, Department of Chemical Engineering, Faculty of Chemical & Natural Resources Engineering, Universiti Teknologi Malaysia, 80990 Johor Bahru, Malaysia

Abstract

This paper discusses the application of artificial neural networks in the area of process monitoring, process control and fault detection. Since chemical process plants are getting more complex and complicated, the need of schemes that can improve process operations is highly demanded. Artificial neural network can provide a generic, non-linear solution, and dynamic relationship between cause and effect variables for complex and non-linear processes. This paper will describe the application of neural network for monitoring reactor temperature, estimation and inferential control of a fatty acid composition in a palm oil fractionation process and detection of reactor sensor failures in the Tennessee Eastman Plant (TEP). The potential for the application of neural network technology in the process industries is great. Its ability to capture and model process dynamics and severe process non-linearities makes it powerful tools for process monitoring, control and fault detection.

Keywords: Artificial neural networks; Process monitoring; Process control; Fault detection

1. Introduction

As chemical process plants are getting more complex and more tightly integrated, the pressure on chemical/process engineers are more increasing. The pressure to improve yield, reduce wastage, improve product quality and above all increase profits makes it essential to increase the efficiency of process operations. In order to achieve this, one possible approach is through improvement of existing process monitoring, process control and fault detection systems.

The application of artificial neural networks (ANNs) to modeling, control and fault detection for non-linear processes has been intensively studied in recent years (Narendra and Parthasarathy, 1990; Hunt et al., 1992; Lightbody and Irwin, 1994). Compared with the conventional polynomial model-based non-linear identification, only the model order and the time delay are needed in neural modeling, as a neural network can represent any non-linearity to any pre-specified accuracy by its topology and non-linear transformation, provided that there are enough neurons in the hidden layer (Funahashi, 1989). Model structure selection should, therefore, be investigated for use in neural modeling.

Many process monitoring, control and fault detection schemes are based upon a representation of the dynamic relationship between cause and effect variables. As with standard linear modeling techniques, ANNs are capable of approximating the dynamic relationships between cause and effect variables. In contrast to linear techniques, ANNs show a potential of being able to capture non-linear relationships.

This paper details some applications of neural networks in chemical/process engineering that have been studied. The paper begins by describing ANNs in more detail. This is followed by applications of neural networks in the area of process monitoring, process control and fault detection in Section 3, 4 and 5 respectively. Finally, a list of conclusions is provided in Section 6.

2. Neural Networks

A neural network consists of a large number of simple processors called neurons. A typical neuron is shown in Fig. 1. The neuron has n inputs, $x_1, x_2, \ldots, x_n$. These inputs can come from other units, or from some external source. The output of the unit y for the neural networks is given as:

* Corresponding author.
E-mail: kamadean@mailcity.com
[a] arshad@fkkksa.utm.my
Tel.: 607-5505410; fax: 607-5581463

$$y = \frac{1}{1+e^{-A}} \ , \qquad (1)$$

where

A = the element activation = $\sum_{i=1}^{n} w_i x_i$,

w_i = a weighting term.

Other types of transfer functions can be used besides the one described in equation (1).

These simple processing units are arranged in layers, as shown in Fig. 2. The networks shown had three layers. Each neuron in the first layer has single input, the external input to the neural network. Each neuron in the second layer has an input from every neuron in the first layer and one additional input with a fixed value of unity. Each neuron in the third layer has an input from every neuron in the second layer and, like the second layer, one additional input with a fixed value of unity. The output of the third layer is the external output of the neural network.

The second layer, which has no direct connections to the external world, is usually referred to as a hidden layer. The first layer is called input layer, while the third layer is called the output layer. More complicated networks can be utilized which have additional hidden layer.

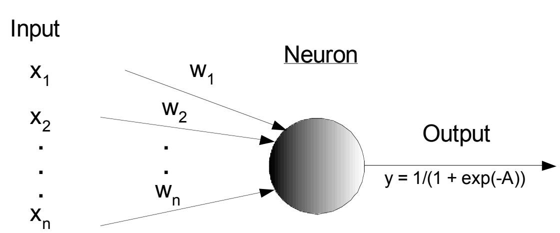

Fig. 1: An individual neural network processing unit (neuron).

The purpose for using the neural network is to obtain a mapping from a vector X to a vector Y. The size of the input and output layers are fixed by the number of components of X and Y, respectively. The user specifies the number of neurons N_h in the hidden layer. For a network with a single hidden layer, and a user-specified hidden layer size, the number of weighting terms in the network is:

$$N_t = N_x N_h + N_h N_y + N_h + N_y , \qquad (2)$$

where

N_t = the total number of weighting terms,
N_x = the dimension of X,
N_y = the dimension of Y, and
N_h = the size of the hidden layer.

For a given external input X to the neural network, the network described above will return an external output value Y:

$$\widehat{Y} = f(X, N_t, w_{i}, _{i=1,N_t}) , \qquad (3)$$

where

$\hat{Y}$ = is the network predicted of Y.

The terminology is used to differentiate the predicted value of $\hat{Y}$ from the observed value of Y associated with the input X. It is desired that the difference between the predicted and observed values be as small as possible. The user can specify the network topology – the number and sizes of the hidden layers – as well as the values of the weighting terms. Usually, the major decision of the user is specification of the network topology. Then, the network weighting terms are found as the solution of the optimization problem. Obtaining the optimal weights for the network is known as training the network. During a training cycle, the network is presented with examples of the type of mapping desired. These examples, called training vectors, consist of two parts, an input X *a*nd a target Y.

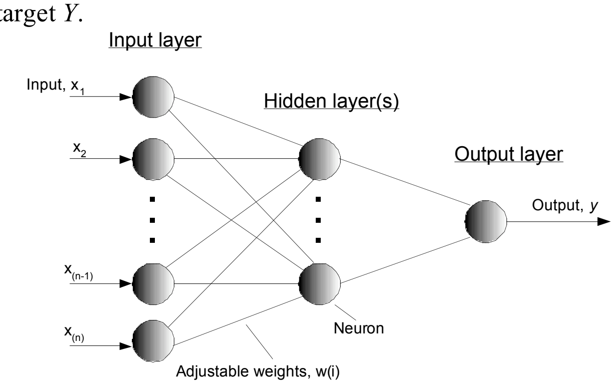

Fig. 2: An example of a three-layer backpropagation neural network. The input vector is of dimension n and the output vector is of dimension 1.

The input is the vector being mapped from and the target is the vector being mapped to. The output of the network is a predicted output $\hat{Y}$. The optimization is set up to minimize the difference between the predicted values of $\hat{Y}$ and the observed values Y. The finding of the set of weights which minimize the error between Y and $\hat{Y}$ is called the training of the network.

2.1 Dynamic process model formulation and identification

The process model for a sampled data system can be formulated in its most general form as:

$$y_t = f(y_{t-1}, y_{t-2}, y_{t-3}, \ldots ,$$

$$u_{t-1}, u_{t-2}, u_{t-3}, ..., e_t, \theta_t, t)\,, \qquad (4)$$

where

f = specified functional relationship,
y_t = process output at time t,
u_t = measured process input at time t,
e_t = unmeasured process input at time t,
t = discrete time index (integer values only), and
θ_t = set of parameter values at time t.

The inputs and outputs can be single (scalar) or multivariable (vector) values.

The identification process for a given set of input-output can be described in three steps:

1. Postulate a structure to the model (f).
2. Calculate the *best* estimate of θ_t.
3. Validate the resulting model.

The identification process is often iterative in nature. For example, the results of the validation step may suggest a better structure f for describing the process.

Dynamic model identification work performed for identification is almost always done assuming a simpler model structure than equation (4). It is usually assumed that the desired model is time-invariant and linear, leading to a model of the form:

$$y_t = \sum_{i=1}^{x} g_i u_{t-1} + e_t\,, \qquad (5)$$

where g_i are the process parameters of the system.

2.2 Neural network training with cross-validation

We wish to find some estimation of θ which allows us to predict values of y. The concept of cross-validation is that after estimation using a given sample of data, the quality of the mapping is evaluated using a different set of data. The best mapping is defined as the one which minimizes the prediction error on a data set for which it was not trained.

The cross validation technique can be summarized as:

1. Separate the M data points collected into two sets, a training set $\{x_i, y_i, i = 1,\ldots, m\}$ and a test set $\{x_i, y_i, i = m + 1,\ldots, M\}$.
2. Construct a number of mappings using the training set.
3. Evaluate the mappings using the test set.
4. Select the mapping which minimizes some criterion applied to the test set.

3. Neural networks for process monitoring

The process system that exhibits non-linear characteristics can be successfully monitored through the utilization of neural networks. Data usually taken from sensors are used as inputs for the network and outputs are the systems that need to be monitored such as temperature, pressure and product quality. Based on the monitoring, the decision is usually made whether the process should continue on, or a control action should be taken to make correction, or faulty condition can be detected. In other words, process monitoring is the starting step before any process control or fault detection action is taken.

To monitor temperature of a reactor, for example, sets of data that have an effect to the reactor temperature are needed. These data are the inputs for the neural network and the output is the temperature that needs to be monitored. The input data contain data of reactor pressure, reactor temperature and cooling water temperature with 1 and 2 delayed terms. Elman network is used in this study with 3 sets of different data needed for training, validation and testing. Fig. 3 shows the predicted of reactor temperature compared to the actual temperature that monitored by Elman network. The result shows that Elman network predict the reactor temperature successfully.

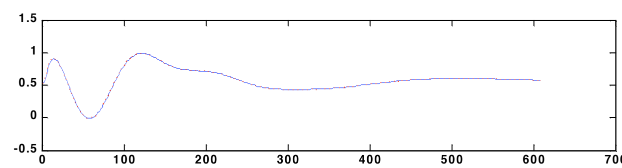

Fig. 3: Neural network predictor for reactor temperature

The use of neural network for process monitoring has given a successful result. This is because the ability of neural networks to capture and model process dynamics and severe process non-linearities.

4. Neural networks for process control

In the neural network-based control systems, a neural network is often trained to estimate the unknown nonlinear process and a controller is then formulated based on the neural network. The formulation of the control signal, however, is not easy as it has to be determined from the neural network model that is nonlinear with respect to its input arguments. It is, therefore, necessary to develop an approach to simplify the control signal formulation. The problem of control signal determination can be regarded as that of the inversion of the neural network model, and it may be solved by nonlinear optimization.

One of the neural network-based control systems that have been studied is inferential control. The idea of inferential control is founded on the use of secondary measurements in the computation of control actions for the control of the primary variables. Fig. 4 displays the schematic diagram of an inferential estimation system. The system uses measurements of secondary process outputs, such as temperatures, pressures and/or flowrates, to infer the effect of unmeasurable disturbances on primary process outputs, in other words, variables being controlled, such as product quality. The control system uses its inference to adjust the control effort to counteract the effect of the unmeasurable disturbances on the product quality. Inferential control can be viewed as an extension of feedforward control which infers the effect of measurable disturbances on the product quality and adjusts the control effort to counteract the effect of the measured disturbances.

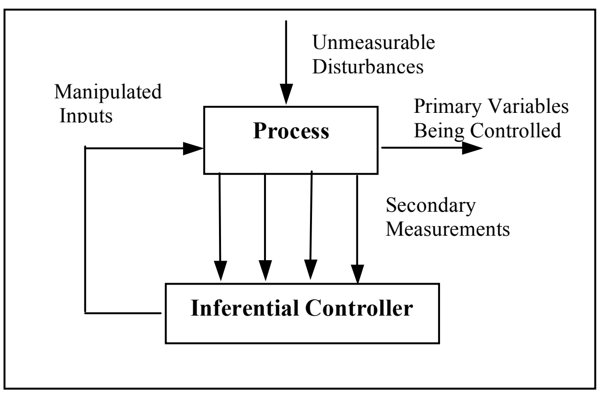

Fig. 4: Inferential Control Strategy

The capability of neural networks in providing inferential measurement of the fatty acid composition in a palm oil fractionation process is illustrated in Fig. 5. As clearly displayed, the neural network model is capable of estimating the product composition accurately and continuously during the operation of the plant.

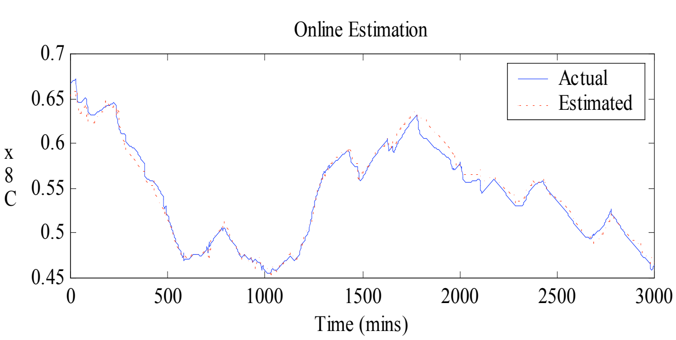

Fig. 5: On-line Estimation of Product Composition Using ANN Model

control loop. Here, estimated values of product composition using the neural network model will be used instead of the actual measurement. The performance of this proposed strategy was compared to the indirect composition control using temperature control (i.e., base case). The inferential strategy was tested on set point tracking of the product composition. As observed in Fig. 6, satisfactory performance was obtained

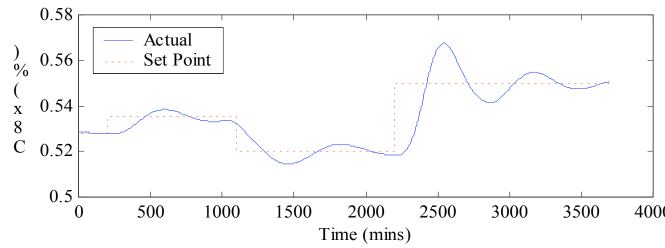

Fig. 6: Set point tracking of product composition using inferential control

The use of inferential estimators constructed using ANN has provided efficient estimation of the product composition. The implementation of the inferential control has also been successful. It is therefore concluded that the use of inferential estimation and control strategy is a viable approach when dealing with frequently disturbed processes or in cases where the feed composition is uncertain.

5. Neural networks for fault detection

Fault detection and diagnosis is becoming extremely important for safe and optimal operation of process plants. There has been considerable work done on fault detection using neural networks. In fault detection application, the inputs to the network include the symptoms present or absent in the system being diagnosed, and the outputs of the network represent the present or absence of particular fault causes. In fault diagnosis of physical systems, such as chemical plants, the inputs to the network include the sensor values or alarm states of the process, either used directly or with minor data conditioning performed first.

Neural networks used for process fault detection generally use sensor measurement and process alarms as inputs, while the outputs represent particular fault types, or categories. In many of fault detection and diagnosis neural network systems previously developed (Hoskins and Himmelblau, 1988; Venkatasubramanian and Chen, 1989; Watanabe et al., 1989), each output neuron corresponds to one particular fault possibility. In the ideal situation, if the value of a neuron in the output layer of the network is equal to one, then the fault represented by that particular neuron is considered to be present. Conversely, if the output of a neuron in the output layer is equal to zero, the fault represented by that particular neuron is judged to be absent. A fault-free state is indicated when the values of all output neurons are equal to zero, and multiple faults are indicated when the values of multiple output neurons are equal to one.

In this paper, detection of reactor sensor failures in the Tennessee Eastman Plant (TEP) has been studied. Fig. 7 shows the schematic diagram of the process.

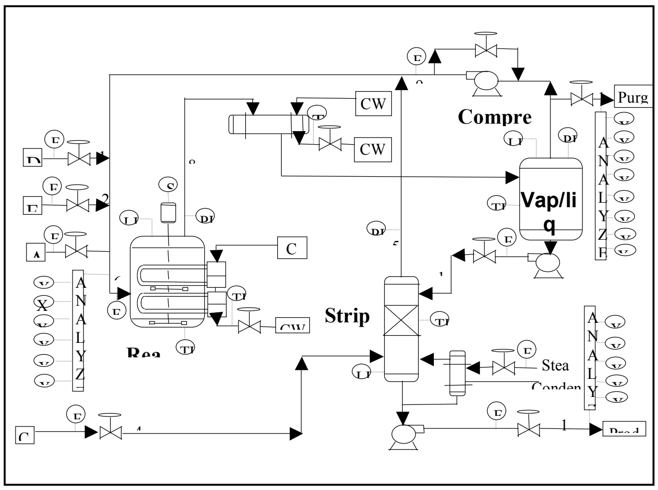

Fig. 7: Schematic diagram of Tennessee Eastman Plant (TEP).

This study focused on the malfunctions of the process caused by the failure of the pressure, temperature and cooling water temperature sensors in the reactor. Faulty conditions are simulated using the Tennessee Eastman Plant model coded in MATLAB language. Sensor failures are created causing the normal process operation to shift to a faulty operation mode. For the pressure sensor, deviation of 4.0% or greater from the normal condition is assumed to cause malfunction to the process. Similarly, for temperature and cooling water temperature sensors the figures are 6.0% and 3.0% respectively.

For fault detection scheme, two types of networks are needed. First is predictor, to predict behavior of reactor temperature and second is classifier, to classify type of fault. Elman network is used for predictor with pressure, temperature and cooling water temperature with 1 and 2 delayed terms as inputs. Behavior of reactor temperature is depicted in Fig. 3. Meanwhile for classifier multilayer feedforward neural network with one input and three outputs F1, F2 and F3 represent pressure, temperature and cooling water temperature sensor fault, respectively. These two networks are trained using Levenberg-Marquardt learning algorithm. Input for classifier is in the form of residual signal from predictor.

The outputs of the classifier are set between the values of 0 and 1. In this study, the classifier is designed in such a way that the faults are monitored and alarm signal is generated when the classifier's output reached the output index 0.8. 0.8 is an assigned value for the residual reactor pressure. When the residual reactor pressure beyond the assigned value, it indicated that the actual reactor pressure has deviated from its normal operating condition. Fig. 8 and 9 show fault detection for reactor pressure sensor fault (F1) and reactor cooling water temperature sensor fault (F3), respectively.

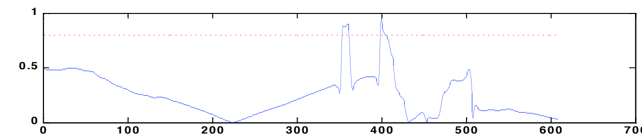

Fig. 8: Fault detection for reactor pressure sensor fault (F1)

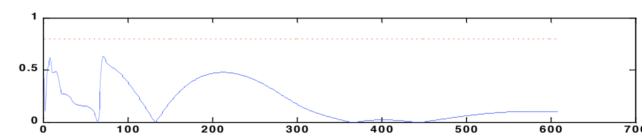

Fig. 9: Fault detection for reactor cooling water temperature sensor fault (F3)

The results revealed the success of the classifiers in detecting the process fault in the dynamic operation condition. The classifier's outputs reached the index 0.8 to indicate the violation of operating limit and the cause of the violation.

6. Conclusion

ANN-based systems for process monitoring, process control and fault detection have been studied. Application of neural network in monitoring reactor temperature gave a successful result. Neural network also used as an estimator and inferential controller to determine composition of fatty acid in a palm oil fractionation process. Its results also indicated that neural network can be successfully applied. For detecting sensor fault, neural network-based classifiers have successfully been applied to detect sensor faults in the Tennessee Eastman Plant. From all the results, it is shown that neural network can capture and model process dynamics and severe process non-linearities makes neural network a powerful tool in process monitoring, control and fault detection.

References

Funahashi, K. I., 1989. On the approximate realization of continuous mappings by neural networks. *Neural Networks* 2, 183-192.

Hoskins J. C. and D. M. Himmelblau, Artificial neural network models of knowledge representation in chemical engineering. *Computers chem.. Engng* 12, 881-890 (1988).

Hunt, K. J., Sbarbaro, D., Zbikowski, R., Gawthrop, P. J., 1992. Neural networks for control systems – a survey. *Automatica* 28 (6), 1083-1112.

Lightbody, G., Irwin, G.W., 1994. Neural network modeling of a polymerization reactor. In: Proceedings of the IEE International Conference Control '94, Coventry, UK, 1, pp. 237-242.

Narendra, K. S., Parthasarathy, K., 1990. Identification and control of dynamic systems using neural networks. *IEEE Trans. Neural Networks* 1, 4-27.

Venkatasubramanian V. and K. Chen, A neural network methodology for process fault diagnosis. *AIChE Jl* 35, 1993-2002 (1989).

Watanabe K., I. Matsuura, M. Abe, M. Kubota and D. M. Himmelblau, Incipient fault diagnosis of chemical processes via artificial neural networks. *AIChE Jl* 35, 1803-1812 (1989).